# Paradigmas de Programación Correspondencia de Curry-Howard

2do cuatrimestre de 2024
Departamento de Computación
Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

# Sistema de tipos para el cálculo Lambda

```
Reglas de tipado
    \Gamma \vdash \mathsf{true} : \mathsf{bool}^{\mathsf{T-TRUE}}
                                                                                                         -T-FALSE

        Γ ⊢ false : bool

    \Gamma \vdash M: bool \Gamma \vdash N : \tau \quad \Gamma \vdash P : \tau
                \Gamma \vdash \text{if } M \text{ then } N \text{ else } P : \tau
                                                            \frac{\Gamma, x : \tau \vdash M : \sigma}{\Gamma \vdash \lambda x : \tau \cdot M : \tau \to \sigma}T-ABS
    \overline{\Gamma, x : \tau \vdash x : \tau}T-VAR
    \Gamma \vdash M : \tau \to \sigma \quad \Gamma \vdash N : \tau
                     \Gamma \vdash MN : \tau
```

# Sistema de tipos para el cálculo Lambda

# Deducción natural

- Ignoremos los términos del lambda cálculo
- Notar que las reglas de tipado se corresponden con reglas de deducción natural:

## Correspondencia de Curry

Observación realizada sobre la lógica combinatoria:

#### Lógica combinatoria

Variante del cálculo lambda que sustituye a las abstracciones por un conjunto limitado de combinadores.

▶ Curry & Feys observaron que si se lee el tipo  $\sigma \to \tau$  como una implicación  $\sigma \Rightarrow \tau$ , luego

la regla de tipado de la aplicación de una función es la regla **modus ponens** 

# Pruebas y Programas

```
Proposiciones \leftrightarrow Tipos
Pruebas \leftrightarrow Términos
```

Un juicio  $\vdash \tau$  es derivable sí y sólo sí el tipo  $\tau$  está habitado, esto es, existe un término M tal que  $\vdash M$ :  $\sigma$  es derivable.

# Ejemplo

#### ¿Es derivable $\vdash \sigma \Rightarrow \sigma$ ?

Si, por ejemplo:

$$\frac{\overline{\sigma \vdash \sigma}^{\mathsf{ax}}}{\vdash \sigma \Rightarrow \sigma} \Rightarrow_{i}$$

Corresponde al siguiente juicio de tipado:

$$\frac{\overline{x : \sigma \vdash x : \sigma}^{\text{T-VAR}}}{\vdash \lambda x : \sigma.x : \sigma \to \sigma}^{\text{T-ABS}}$$

El **término**  $\lambda x : \sigma.x$  se asocia con la **prueba** de  $\sigma \Rightarrow \sigma$  que se muestra en la parte superior

## Ejemplo

#### ¿Es derivable $\vdash \sigma \Rightarrow \sigma$ ?

También existe la siguiente prueba

$$\frac{\overline{\sigma \Rightarrow \sigma \vdash \sigma \Rightarrow \sigma}^{\mathsf{ax}}}{\vdash (\sigma \Rightarrow \sigma) \Rightarrow \sigma \Rightarrow \sigma} \Rightarrow_{i} \frac{\overline{\sigma \vdash \sigma}^{\mathsf{ax}}}{\vdash \sigma \Rightarrow \sigma} \Rightarrow_{i}$$

$$\vdash \sigma \Rightarrow \sigma$$

Corresponde al siguiente juicio de tipado:

$$\frac{\frac{}{x:\sigma \to \sigma \vdash x:\sigma \to \sigma}^{\text{T-VAR}}}{\vdash \lambda x:\sigma \to \sigma.x:(\sigma \to \sigma) \to \sigma \to \sigma}^{\text{T-ABS}} \quad \frac{\frac{}{y:\sigma \vdash y:\sigma}^{\text{T-VAR}}}{\vdash \lambda y:\sigma.y:\sigma \to \sigma}^{\text{T-ABS}}}{\vdash (\lambda x:\sigma \to \sigma.x)(\lambda y:\sigma.y):\sigma \to \sigma}^{\text{T-APP}}$$

El **término**  $(\lambda x : \sigma \to \sigma.x)(\lambda y : \sigma.y)$  se asocia con la **prueba** que se muestra en la parte superior.

## Ejemplo

#### ; Es derivable $\vdash \sigma \Rightarrow \sigma$ ?

También existe la siguiente prueba

$$\frac{\frac{\overline{\sigma \vdash \sigma} ax}{\sigma \vdash \sigma \Rightarrow \sigma} \Rightarrow_{i} \quad \overline{\sigma \vdash \sigma} ax}{\frac{\sigma \vdash \sigma}{\vdash \sigma \Rightarrow \sigma}} \Rightarrow_{e} = \frac{\overline{\sigma} \vdash \sigma}{} \Rightarrow_{e} = \frac{\overline{\sigma} \vdash$$

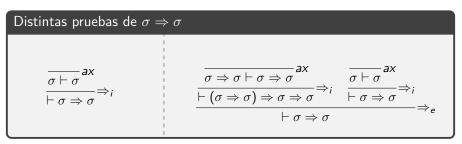
$$\frac{\overline{x : \sigma, y : \sigma \vdash y : \sigma}^{\text{T-VAR}}}{x : \sigma \vdash \lambda y : \sigma . y : \sigma \to \sigma}^{\text{T-VAR}} \xrightarrow{x : \sigma \vdash x : \sigma}^{\text{T-VAR}} \xrightarrow{\text{T-APP}}$$

 $\vdash \lambda x : \sigma.(\lambda y : \sigma.y)x : \sigma \to \sigma$ 

El **término**  $\lambda x : \sigma.(\lambda y : \sigma.y)x$  se asocia con la **prueba** que se muestra en la parte superior.

#### Pruebas vs términos

- Una fórmula puede tener muchas pruebas distintas.
- Distintas pruebas corresponden a distintos juicios de tipado, es decir distintos términos.
- Notar que algunas pruebas de la misma proposición son mas complejas que otras:



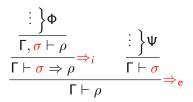
## Correspondencia de Curry-Howard

- William Alvin Howard extiende la correspondencia:
  - Tratando los restantes conectivos lógicos.
  - Usando el cálculo lambda en lugar de la lógica combinatoria.
  - Mostrando una correspondencia entre la simplificación de pruebas y la computación.

# Simplificación de pruebas

#### Corte (Cut)

Un corte es una afirmación intermedia (un lema) que probamos a pesar de que no es una subfórmula de la afirmación final (el teorema)



#### $\sigma$ es un corte

Caracterizado por el uso de  $\Rightarrow_i$  seguido por  $\Rightarrow_e$ 

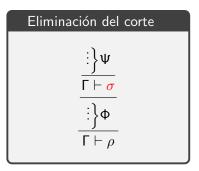
# Simplificación de pruebas

#### Eliminación de Corte (Cut)

Reescribir una prueba de manera tal que no tenga cortes:

▶ Eliminamos  $\sigma$  reemplazando cada uso  $\sigma$  en la prueba de  $\rho$  por una copia de la prueba de  $\sigma$ .

$$\frac{\frac{\vdots}{\Gamma, \sigma \vdash \rho}}{\frac{\Gamma \vdash \sigma \Rightarrow \rho}{\Gamma \vdash \rho}} \Rightarrow_{i} \frac{\vdots}{\Gamma \vdash \sigma} \Rightarrow_{e}$$



# Eliminación de corte : Ejemplo

Eliminación de corte
$$\frac{\begin{cases} }{\sigma \Rightarrow \sigma \vdash \sigma \Rightarrow \sigma} ax \\ \hline{\vdash (\sigma \Rightarrow \sigma) \Rightarrow \sigma \Rightarrow \sigma} \Rightarrow_{i} \\ \vdash \sigma \Rightarrow \sigma \end{cases} \xrightarrow{\sigma \vdash \sigma} ax \end{cases} \Psi$$

$$\frac{}{\vdash (\sigma \Rightarrow \sigma) \Rightarrow \sigma \Rightarrow \sigma} \Rightarrow_{i} \xrightarrow{\vdash \sigma \Rightarrow \sigma} \Rightarrow_{i} \\ \vdash \sigma \Rightarrow \sigma \Rightarrow_{e} \xrightarrow{} \Rightarrow_{e}$$

# Computación como simplificación de pruebas

#### Eliminación de corte y reducción $\beta$

Un paso de reducción  $\beta$  (esto es, aplicar  $\operatorname{E-APPABS})$  se corresponde con una eliminación de corte.

$$\frac{\vdots}{\Gamma, \sigma \vdash M : \rho} \qquad \qquad \vdots \qquad \qquad \frac{\vdots}{\Gamma \vdash N : \sigma} \\
\frac{\Gamma \vdash \lambda x : \sigma . M : \sigma \to \rho}{\Gamma \vdash (\lambda x : \sigma . M) N : \rho} \qquad \qquad \frac{\Gamma \vdash N : \sigma}{\Gamma \vdash M \{x := N\} : \rho}$$

#### Normalización

#### Forma normal

Una prueba está en forma normal si no posee cortes.

## Theorem (Normalización de pruebas)

Toda prueba puede ser "normalizada" mediante la eliminación sucesiva de cortes.

## Conjunción

#### Extendemos la sintaxis

$$\sigma, \tau, \dots ::= \dots \mid \sigma \times \tau$$
  
 $M, N, \dots ::= \dots \mid \langle M, N \rangle \mid \text{fst } M \mid \text{snd } N$ 

$$\frac{\Gamma \vdash \sigma \quad \Gamma \vdash \tau}{\sigma \vdash \sigma \land \tau} \land_{i}$$

$$\frac{\Gamma \vdash \sigma \land \tau}{\Gamma \vdash \sigma} \land_{e_{1}} \frac{\Gamma \vdash \sigma \land \tau}{\Gamma \vdash \tau} \land_{e_{2}}$$

#### Producto

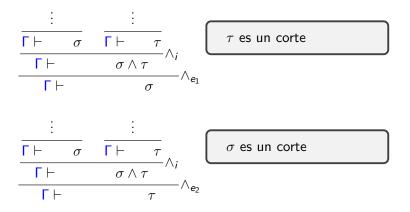
#### Extendemos la sintaxis

$$\begin{array}{lll} \sigma, \tau, \ldots & ::= \ldots \mid \sigma \times \tau \\ M, N, \ldots & ::= \ldots \mid \langle M, N \rangle \mid \text{fst } M \mid \text{snd } N \end{array}$$

$$\frac{\Gamma \vdash M : \sigma \qquad \Gamma \vdash N : \tau}{\sigma \vdash \langle M, N \rangle : \sigma \times \tau}$$

$$\frac{\Gamma \vdash M : \sigma \times \tau}{\Gamma \vdash \mathsf{fst} \ M : \sigma} \qquad \frac{\Gamma \vdash M : \sigma \times \tau}{\Gamma \vdash \mathsf{snd} \ M : \tau}$$

# Conjunción : Corte



# Conjunción: Eliminación de corte

## Producto: Reducción

$$\begin{array}{c|c} \vdots & \vdots \\ \hline {\Gamma \vdash M : \sigma} & \overline{\Gamma \vdash N : \tau} \\ \hline {\Gamma \vdash \langle M, N \rangle : \sigma \times \tau} \\ \hline {\Gamma \vdash \operatorname{fst} \langle M, N \rangle : \sigma} \\ \hline \vdots & \vdots \\ \hline {\Gamma \vdash M : \sigma} & \overline{\Gamma \vdash N : \tau} \\ \hline {\Gamma \vdash \langle M, N \rangle : \sigma \times \tau} \\ \hline {\Gamma \vdash \operatorname{snd} \langle M, N \rangle : \tau} \\ \hline \end{array} \rightarrow \begin{array}{c} \vdots \\ \hline {\Gamma \vdash N : \tau} \\ \hline \end{array}$$

# Disjunción

#### Extendemos la sintaxis

$$\begin{array}{lll} \sigma, \tau, \dots & ::= & \dots & \mid \sigma + \tau \\ M, N, P, \dots & ::= & \dots & \mid \mathsf{left}^\sigma \; M \mid \mathsf{right}^\sigma \; M \\ & \mid \mathsf{case} \; M \; \mathsf{with} \{\mathsf{left} \; x \to N, \mathsf{right} \; x \to P\} \end{array}$$

#### Suma

#### Extendemos la sintaxis

```
\begin{array}{lll} \sigma, \tau, \dots & & ::= & \dots & \mid \sigma + \tau \\ M, N, P, \dots & ::= & \dots & \mid \mathsf{left}^\sigma \; M \mid \mathsf{right}^\sigma \; M \\ & & \mid \mathsf{case} \; M \; \mathsf{with} \{\mathsf{left} \; x \to N, \mathsf{right} \; x \to P\} \end{array}
```

$$\frac{\Gamma \vdash M : \sigma}{\Gamma \vdash \mathsf{left}^{\tau} \; M : \sigma + \tau} \qquad \frac{\Gamma \vdash M : \tau}{\Gamma \vdash \mathsf{right}^{\sigma} \; M : \sigma + \tau}$$

$$\frac{\Gamma \vdash M : \sigma + \tau \qquad \Gamma, x : \sigma \vdash M : \rho \qquad \Gamma, x : \sigma \vdash N : \rho}{\Gamma \vdash \mathsf{case} \; M \; \mathsf{with} \{\mathsf{left} \; x \to N, \mathsf{right} \; x \to P\} : \rho}$$

## Disjunción : Corte

```
\forall_{i_1} seguido de \forall_e es un corte
\frac{\Gamma \vdash M : \sigma}{\Gamma \vdash \mathsf{left}^{\tau} \ M : \sigma + \tau} \lor_{i_1} \qquad \frac{\Gamma, x : \sigma \vdash N : \rho}{\Gamma, x : \tau \vdash P : \rho} \lor_{e}
            \Gamma \vdash \mathsf{case} \; \mathsf{left}^{\tau} \; M \; \mathsf{with} \{ \mathsf{left} \; x \to N, \; \mathsf{right} \; x \to P \} : \rho
                                                                                                                                              \bigvee_{i_2} seguido de \bigvee_e es un corte
\frac{\Gamma \vdash \mathsf{right}^{\sigma} \ M : \sigma + \tau}{\Gamma \vdash \mathsf{right}^{\sigma} \ M : \sigma + \tau} \vee_{i_{2}} \qquad \frac{\Gamma, x : \sigma \vdash N : \rho}{\Gamma, x : \tau \vdash P : \rho} \vee_{e}
             \Gamma \vdash \mathsf{case} \ \mathsf{right}^{\tau} \ M \ \mathsf{with} \{\mathsf{left} \ x \to N, \mathsf{right} \ x \to P\} : \rho
```

# Suma: Reducción (1)

```
 \begin{array}{c|cccc} \overline{\Gamma \vdash M : \sigma} & \vdots & \vdots \\ \hline \overline{\Gamma \vdash \mathsf{left}^{\tau} \ M : \sigma + \tau}^{\bigvee_{i_1}} & \overline{\Gamma, x : \sigma \vdash N : \rho} & \overline{\Gamma, x : \tau \vdash P : \rho} \\ \hline \Gamma \vdash \mathsf{case} \ \mathsf{left}^{\tau} \ M \ \mathsf{with} \{\mathsf{left} \ x \to N, \mathsf{right} \ x \to P\} : \rho \end{array} \\ \vee_e 
                                                                                                                                                                                                                                                                                                              \Gamma \vdash N\{x := M\} : \rho
```

# Suma: Reducción (2)

```
\frac{\overline{\Gamma \vdash M : \tau}}{\Gamma \vdash \mathsf{right}^{\sigma} \ M : \sigma + \tau} \bigvee_{i_2} \quad \frac{\vdots}{\Gamma, x : \sigma \vdash N : \rho} \quad \frac{\vdots}{\Gamma, x : \tau \vdash P : \rho} \bigvee_{e}
             \Gamma \vdash \mathsf{case} \; \mathsf{right}^{\tau} \; M \; \mathsf{with} \{\mathsf{left} \; x \to N, \mathsf{right} \; x \to P\} : \rho
                                                                                                                                                                            \Gamma \vdash M : \sigma
                                                                                                                                                              \Gamma \vdash P\{x := M\} : \rho
```

## Absurdo

#### Extendemos la sintaxis

$$\sigma, \tau, \dots$$
 ::= ... |  $\bot$   
 $M, N, P, \dots$  ::= ... | case  $M$  with $\{\}$ 

$$\frac{\Gamma \vdash \qquad \bot}{\Gamma \vdash \qquad \qquad \sigma} \bot_{e}$$

#### Absurdo

#### Extendemos la sintaxis

```
\sigma, \tau, \dots ::= ... | \bot

M, N, P, \dots ::= ... | case M with\{\}
```

```
\frac{\Gamma \vdash M : \bot}{\Gamma \vdash \mathsf{case}\ M\ \mathsf{with}\{\} : \sigma} \bot_{\mathsf{e}}
```

- ▶ Notar que no hay constructores para el tipo  $\bot$ .
- ► El tipo ⊥ (Void) es el tipo vacío.
- Se puede definir como un tipo de dato algebraico sin constructores.

# Correspondencia de Curry-Howard

## Theorem (Correspondencia de Curry-Howard)

 $A_1, \ldots, A_n \vdash \sigma$  es derivable en NJ ssi existe un término M donde  $fv(M) \subseteq \{x_1, \ldots, x_n\}$  tal que  $x_1 : A_1, \ldots, x_n : A_n \vdash M : \sigma$ .

# Consistencia de la lógica

La relación entre reducción y pruebas permite concluir que la lógica es consistente.

#### Corollary

 $\forall \perp$  (en NJ).

Se obtiene a partir del siguiente razonamiento:

- ▶ Debe existir M, tal que  $\vdash M : \bot$ .
- Por terminación y preservación de tipos, debería existir un valor V, tal que ⊢ V : ⊥. Por analisis de casos en los posibles valores, se puede concluir que no existe.

# Sobre la negación

La negación se puede codificar como:

$$\neg \sigma \equiv \sigma \rightarrow \bot$$

- Notar que la regla:
  - ightharpoonup  $\neg_i$  corresponde  $a \Rightarrow_i$
  - ightharpoonup  $\neg_e$  corresponde  $a \Rightarrow_e$
- De esta manera no hay necesidad de extender al sistema de tipos

#### Tipo Unit

- Se puede considerar que la lógica está extendida con la fórmula ⊤ (fórmula válida).
- Se considera NJ extendido con la siguiente regla:

$$\overline{\Gamma \vdash \top}^{\top_i}$$

► En el cálculo lambda extendemos la sintaxis con el tipo ⊤ que tiene un único elemento.

$$\sigma, \tau, \dots$$
 ::= ... |  $\top$   $M, N, P, \dots$  ::= ... |  $\top$ 

▶ Una única regla de tipado (que se corresponde con  $\top_i$ )

$$\overline{\Gamma \vdash \top : \top}^{\text{T-UNIT}}$$

ightharpoonup El tipo op es un tipo algebraico con un único constructor op.

#### Sobre los booleanos

Los ignoramos porque se pueden codificar.

#### Booleanos como sumas

```
\begin{aligned} \mathsf{Bool} &\equiv \top + \top \\ \mathsf{true} &\equiv \mathsf{left}^\top \\ \mathsf{false} &\equiv \mathsf{right}^\top \\ \mathsf{if} \ \mathit{M} \ \mathsf{then} \ \mathit{N} \ \mathsf{else} \ \mathit{P} &\equiv \mathsf{case} \ \mathit{M} \ \mathsf{with} \{ \mathsf{left}^\top \ \_ \to \mathit{N}, \mathsf{right}^\top \ \_ \to \mathit{P} \} \end{aligned}
```

 Existen codificaciones en el fragmento implicativo (booleanos de Church)

#### Recursión

Extendemos la sintaxis con un nuevo operador

$$M ::= \dots \mid \text{fix } M$$

No se precisan nuevos tipos pero sí una regla de tipado.

$$\frac{\Gamma \vdash M : \sigma \to \sigma}{\Gamma \vdash \mathsf{fix} \ M : \sigma} \mathsf{T\text{-}FIX}$$

# Semántica operacional small-step

No hay valores nuevos pero sí reglas de evaluación nuevas.

$$\frac{M \to M'}{\text{fix } M \to \text{fix } M'} \text{E-FIX}$$

$$\overline{\text{fix } (\lambda x : \sigma.M)} \rightarrow M\{x := \text{fix } (\lambda x : \sigma.M)\}$$
 E-FIXBETA

# **Ejemplos**

```
Sea M el término
```

```
\lambda f: \mathsf{nat} \to \mathsf{nat}.

\lambda x: \mathsf{nat}.

if \mathsf{iszero}(x) then \underline{1} else x * f(\mathsf{pred}(x))
```

en

fix  $M \underline{3}$ 

# **Ejemplos**

Ahora podemos definir funciones parciales:

fix 
$$(\lambda x : \sigma.x)$$

- Notar que  $\vdash$  fix  $(\lambda x : \sigma.x) : \sigma$  para cualquier  $\sigma$ .
- ▶ En particular, vale para  $\sigma = \bot$ .
- $\blacktriangleright$  En consecuencia, si se extiende NJ con un operador fix , la lógica sería inconsistente ( $\vdash \bot$  sería derivable)